(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 0 343 683 B1

(12) EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention of the grant of the patent:
03.04.1996 Bulletin 1996/14

(51) Int Cl.6: G06F 7/22

(21) Application number: 89109538.2

(22) Date of filing: 26.05.1989

(54) Control data generator for sort processor

Steuerdatengenerator für Sortierprozessor

Générateur de données de commande pour un processeur de triage

(84) Designated Contracting States:
DE FR GB

(30) Priority: 26.05.1988 JP 129023/88

(43) Date of publication of application:
29.11.1989 Bulletin 1989/48

(73) Proprietors:
- MITSUBISHI DENKI KABUSHIKI KAISHA
Chiyoda-ku Tokyo (JP)
- Kitsuregawa, Masaru
Matsudo-shi Chiba-ken (JP)

(72) Inventors:
- Kitsuregawa, Masaru
Matsudo-shi Chiba-ken (JP)
- Fushimi, Shinya
Kamimachiya Kamakura-shi (JP)

(74) Representative: Goddar, Heinz J., Dr. et al
FORRESTER & BOEHMERT
Franz-Joseph-Strasse 38
D-80801 München (DE)

(56) References cited:
- 'Proceedings 5th International Workshop on Database Machines, Karuizawa, Japan, October 5-8, 1987' 1987 pages 144-157, M.Kitsuregawa et al.: " Design and Implementation of High Speed Pipeline Merge Sorter with Run Length Tuning Mechanism"

Printed by Jouve, 75001 PARIS (FR)

Description

The invention relates to a high speed pipeline merge sorter with a run length tuning mechanism.

A conventional control data generator for a sort processor such as shown in FIG. 1 of the Proceedings of the Japanese Electronic Communications Society, J66-D, March 1983, p 333, is shown in FIG. 6. The control data generator includes four sort processors 1-4, four memories 5-8 each corresponding to each sort processor, and an indicator 9 for the length (L) of data set by the sort processor. The i-th sort processor from left is provided with a memory which has a capacity of $2^{i-l}L$.

In operation, the data to be sorted is sequentially input into the sort processor 1. The sort processor 1 stores one of the input data in the memory 5 and compares it with the next input data. For sorting in descending order, the sort processor 1 first outputs the greater one between the two data and then the smaller one to the next sort processor 2. This operation is repeated in the sort processor 1.

The sort processor 2 stores the descending order 2 data set in the memory 6, in which the two data are sorted in descending order, and merges it with the next input descending order 2-data set to output a descending order 4 data set to the sort processor 3. This operation is repeated in the sort processor 2.

Generally, the i-th sort processor merges the descending order 2^{i-1}-data set sent from the (i-1)th sort processor to output a descending order 2^i -data set to the (i+1)th sort processor. This operation is repeated in the i-th sort processor. Thus, when N (= 2^n) data are input, the n-th sort processor outputs a descending order N-data set sorted.

The above operation, however, is possible only if the length (X) of data to he sorted is either equal to or less than the length (L) of data set by the sort processor.

In accordance with the Proceedings of the 5th International Workshop on Database Machines, October 1986, p 144, even if the input data length (X) is greater than the set data length (L), it is possible to sort a great number of such data by dynamically switching the first sort processor 1 between the MRG mode in which merge is carried out as described above and the NM mode in which no merging operation is performed, with the input data directly transmitted to the sort processor 2, depending on the input data length (X) to thereby use the memories 5-8 more efficiently.

For example, let X = 1.1 L in the descending order sorting. Since the capacity of the memory 8 for the sort processor 4 is 8 L, it is possible to store seven data in the memory as follows:

8 L/X = 8 L/1.1 L = 7.27... 7 (integer part). The sort processor 3 outputs a descending order 7-data set having a data length of 7 to the sort processor 4 and repeats this operation. The sort processor 2 repeats outputting a descending order 3-data set and a descending order 4-data set to the sort processor 3. The sort processor 1 repeats outputting descending order 1-data, 2 data, 2-data, and 2-data sets to the sort processor 2. These operations are shown in FIG. 5.

In FIG. 5, the data 17-30 to be input to the sort processor are shown in the inputting order. Symbols NM 31 and MRG 32-37 indicate that the sort processor 1 is brought into control in the NM mode for data 17 and 24 and into the MRG mode for data 18, 25, etc., respectively. The value (n) of a descending order n-data set to be generated by the sort processor 3 is shown at 10. Similarly, the values (n) of descending order n-data sets to he generated by the sort processor 2 are shown at 11 and 12. The values (n) of descending order n-data sets to be generated periodically by the sort processor 1 are shown at 13-16. The sort processors 1, 2, and 3 in FIG. 6 are represented by P_1, P_2, and P_3.

In this way, the third sort processor 3 always outputs a descending order 7-data set to the fourth sort processor 4. Thus, with the control data generator for a sort processor of FIG. 6, it is possible to sort 14 data under the above conditions. However, the procedure of generating the control data is so complex that there are no ways but by programming. As a result, the sorting capability has been impaired.

Summary of the Invention

Accordingly it is an object of the invention to provide a high speed pipeline merge sorter with a run length tuning mechanism, which requires no programmes for generating the controll data and is made of simple and inexpensive hardware to efficiently generate the control data, thus providing the increase sorting capability.

This object is solved by a sorter having the features of the single patent claim.

The above and other objects, features and advantages of the invention will become more apparent from the following description when taken in conjunction with the accompanying drawings.

Brief Description of the Drawings

FIG. 1 is a block diagram of a control data generator for a sort processor according to an embodiment of the invention;

FIG. 2 is a block diagram of a control data output unit for use in the control data generator of FIG. 1;

FIG. 3 illustrates how to invert the bit order by the hit order inverter of FIG. 2;

FIG. 4 illustrates how to generate and process control data according to another embodiment of the invention;

FIG. 5 illustrates how to generate and process control data according to the prior art and the embodiment of FIG. 1; and

FIG. 6 is a block diagram of a conventional control data generator for a sort processor.

Description of the Preferred Embodiment

In FIG. 1, like reference numerals designate like or corresponding parts of FIG. 6. This control data generator further includes a control data output unit G.

In FIG. 2, the control data output unit G includes a pair of latch registers A and B for storing information about the length of data input from a disk device, etc., a counter C for counting the output of the latch register A, a bit order inverter D for converting the most significant bit to the least significant bit of an output from the counter C, the second MSB to the second LSB, and so on as shown in FIG. 3, a comparator E for comparing the output of the bit order inverter D with the output of the latch register B, and a control Data generating unit F for generating control data based on the output of the comparator E.

The control data is used for switching between the first function or NM mode in which the input data is stored in the corresponding memory outputted without merging and the second function or MRG mode in which the input data are merged in the specific order and outputted from the corresponding memory for later use. Each sort processors 1-4 has both the NM mode and the MRG mode. The comparator E of FIG. 2 is designed to output a logic 0 when a signal from the bit order inverter D is greater than a signal from the latch register B and otherwise a logic 1.

In operation, when the input data length (X) from a disk device, etc. is greater than the set data length (L) in the sort processer (X>L), a d-th sort processor from left in FIG. 1 is selected to compute a value of (2^d L/X), (2^e - 1) and α are stored in the latch registers A and B, respectively, wherein (2^d L/X] = 2^e + ($0 < \alpha < 2^e$).

Then, the control data output unit G starts generating control data.

When the control data is generated, the data is transferred from the latch register A to the counter C for counting. The data from the counter C is inverted in bit order by the bit order inverter D and then compared in the comparator E with the data from the latch register B. The control data generating circuit F outputs control data indicative of the NM mode if the comparison result is a logic 1 and control data indicative of the MRG mode for the present and next input data if the comparison result is a logic 0.

When the inputting operation for the data is completed, the content of the counter C is decreased by 1. When this content becomes negative, it is transferred from latch register A to the counter C. A similar process is repeated for recorded data from a disk device, etc. The control data from the control data generating circuit F is transferred to the selected sort processor so that the sort processor operates according to the control data. Hence, the data input from a disk device, etc. is sorted by the sort processor and transmitted to a host computer, etc.

Although the values 13-16 of FIG. 5 are discussed in the above embodiment, an imaginary sort processor P_0 may be provided to generate the corresponding values 38-45 as shown in FIG. 4. In this case, 2^f and α are stored in the latch registers A and B, respectively, wherein [2^d L/X] = 2^f - α ($0 < \alpha < 2^f$). The counter C is an increment counter with the initial value of 0. The comparator E is modified to output a logic 1 if a value of the data from the bit order inverter D is less than a value of the data from the latch register B and otherwise a logic 0. The control data generating circuit F outputs control data about the NM mode if the data of the comparator E is a logic 1 and otherwise control data about the MRG mode. The counter C is incremented by 1 when the process of each data is completed and if the comparator E outputs a logic 1, and rest to 0 if the result becomes greater than a value of the latch register A. The above configuration produces the same results as those of FIG. 5.

Alternatively, the control data output unit G may be connected to only the first sort processor 1 in FIG. 1, and control data may be sequentially transferred from the first sort processor 1 to the second sort processor 2 to the third sort processor 3 along with the input data to thereby simplify the connection between the control data generating unit and the sort processor.

As has been described above, in accordance with the invention, the control data generator for a sort processor includes a control data output unit which consists of a memory for storing information about the input data length and a control data generation circuit for generating control data to make switching between the first and second functions based on the information from the memory, whereby control data is generated efficiently by the simple and inexpensive hardware without using any control data generation program, thus providing the enhanced sorting capability regardless of the data length.

Claims

1. A high speed pipeline merge sorter with a run length tuning mechanism, which comprises:

   - a plurality of memories (5, 6, 7, 8) for storing input data,

   - a plurality of sort processors (1, 2, 3, 4) each connected to each said memory and having a first function (NM) for storing in and outputting from each said memory input data without merging and a second function (MRG) for merging input data in a specified order and outputting said merged input data from each said memory,

   - a control data generating means (G) connected to said sort processors (1, 2, 3, 4) for generating control data depending on information about the lengths of said input data and thereby switching said sort processor between said first and second function,
   characterized in that said control data genera-

tion means (G) comprises:

- first and second latch registers (A) and (B) for storing said information about the lengths of said input data, whereby the integer value stored in the first latch register (A) is $2^e - 1$ and the integer value stored in the second latch register (B) is α_1 obtained from the integer value $[2^d L/X] = 2^e + \alpha$ by the d-th sort processor, and where L represents the length of a data set by the merge sorter and X the length of input data supplied from outside and $0< \alpha <2^e$,

- a counter (C) connected to said first latch register (A) for counting data of said first latch register,

- a bit order inverter (D) connected to said counter (C) for converting at least the most significant bit into the least significant bit of an output from said counter (C), and

- a comparator (E) connected to said bit order inverter (D) and said second latch register (B) for comparing an output of said bit order inverter with an output of said second latch register (B) to output a logic value to set said sort processor in said first function (NM) when the value in said first latch register (A) is equal to or greater than that of said second latch register (B) and in said second function (MRG) when the value in said first latch register (A) is less than that of said second latch register (B).

Patentansprüche

1. Hochgeschwindigkeits-Mehrfach-blisch-sortierer mit einer Lauflängeabstimmeinrichtung, die umfaßt:

   - eine Vielzahl von Speichern (5, 6, 7, 8) zum Speichern von Eingabedaten,

   - eine Vielzahl von Sortierprozessoren (1, 2, 3, 4), wobei jeder mit jedem der Speicher verbunden ist und eine erste Funktion (NM) besitzt zum Einspeichern und Ausgeben von Eingabedaten aus jedem dieser Speicher ohne Zusammenfügen und eine zweite Funktion (MRG) zum Zusammenfügen der Eingabedaten in vorgewählter Reihenfolge und Ausgeben der zusammengefügten Eingabedaten aus jedem der Speicher,

   - einem Steuerdaten erzeugenden Mittel (G), das mit den Sortierprozessoren (1, 2, 3, 4) verbunden ist zum Erzeugen von Steuerdaten auf Grund von Information über die Länge der Eingabedaten und zum dadurch bewirkten Schalten des Sortierprozessors zwischen der ersten und der zweiten Funktion,

   dadurch gekennzeichnet, daß das Steuerdaten erzeugende Mittel (G) umfasst:

   - erste und zweite Zwischenspeicherregister (A) und (B) zum Einspeichern der Information über die Länge der Eingabedaten, wodurch der ganzzahlige Wert, der in dem ersten Zwischenspeicherregister (A) gespeichert ist, $2^e - 1$ ist, und der ganzzahlige Wert, der in einem zweiten Zwischenspeicherregister (B) gespeichert ist, α ist, erhalten aus dem ganzzahligen Wert $[2^d L/X] = 2^e + \alpha$ durch den d-ten Sortierprozessor, und wobei L die Länge eines Datensatzes vom Mischsortierer darstellt und X die Länge der Eingabedaten, die von außerhalb zugeführt werden, wobei $0 < \alpha < 2^e$,

   - einen Zähler (C), der mit dem ersten Zwischenspeicherregister (A) zum Speichern von Daten des ersten Zwischenspeicherregisters verbunden ist,

   - einem Bit-Reihenfolge-Invertierer (D), der mit dem Zähler (C) gekoppelt ist zum Konvertieren wenigstens des meist-signifikanten Bits (MSB) in das niedrigst-signifikante Bit (LSB) einer Ausgabe aus dem Speicher (C), und

   - einem Komparator (E), der mit dem Bit-Reihenfolge-Invertierer (D) verbunden ist und mit dem zweiten Zwischenspeicherregister (B) zum vergleichen einer Ausgabe aus dem Bit-Reihenfolge-Invertierer mit einer Ausgabe aus dem zweiten Zwischenspeicherregister (B), um einen logischen Wert auszugeben, um den Sortierprozessor in die erste Funktion (NM) zu überführen, wenn der Wert in dem ersten Zwischenspeicherregister (A) gleich oder größer als der in dem zweiten Zwischenspeicherregister (B) ist, und in die zweite Funktion (MRG), wenn der Wert in dem ersten Zwischenspeicherregister (A) geringer als der des zweiten Zwischenspeicherregisters (B) ist.

Revendications

1. Trieur à interclassement en cascade à haute vitesse ayant un mécanisme de syntonisation de format, comprenant

   - une pluralité de mémoires (5, 6, 7, 8) pour stocker des données d'entrée,

   - une pluralité de processeurs de triage (1, 2, 3,

4) respectivement connectés auxdites mémoires et ayant une première fonction (NM) pour stocker dans et émettre depuis chaque dite mémoire des données d'entrée, sans interclassement, et deuxième fonction (MRG) pour interclasser les données d'entrée dans un ordre spécifié et émettre lesdites données d'entrée interclassées depuis chaque dite mémoire,

- un moyen de production de données de commande (G) connecté auxdits processeurs de triage (1, 2, 3, 4), pour produire des données de commande en fonction d'informations concernant les longueurs desdites données d'entrée et de manière à commuter ledit processeur de triage entre lesdites première et deuxième fonctions,
caractérisé en ce que ledit moyen de production de données de commande (G) comprend :

- des premier et deuxième registres de mémorisation (A) et (B) pour stocker lesdites informations concernant les longueurs desdites données d'entrée, de manière que la valeur de nombre entier stockée dans le premier registre de mémorisation (A) soit 2^e -1 et la valeur d'un nombre entier stockée dans le deuxième registre de mémorisation (B) soit α_1, obtenu à partir de la valeur d'un nombre entier $[2^d L/X] = 2^e + \alpha$, par le $d^{ième}$ processeur de triage, et dans lequel L représente la longueur de donnée établie par le trieur à interclassement et X la longueur de données d'entrée fournies depuis l'extérieur et $0 < \alpha < 2^e$,

- un compteur (C) connecté audit premier registre de mémorisation (A) pour compter les données dudit premier registre de mémorisation,

- un inverseur d'ordre de bits (D) connecté audit compteur (C) pour convertir au moins le bit de poids le plus fort en le bit de poids le plus faible d'un signal de sortie provenant dudit compteur (C) et

- un comparateur (E) connecté audit inverseur d'ordre de bits (D) et audit deuxième registre de mémorisation (B), pour comparer un signal de sortie dudit inverseur d'ordre de bits à un signal de sortie du deuxième registre de mémorisation (B), pour produire une valeur logique afin de placer ledit processeur de triage dans ladite première fonction (NM), lorsque la valeur dudit premier registre de mémorisation (A) est supérieure ou égale à celle dudit deuxième registre de mémorisation (B), et dans ladite deuxième fonction (MRG), lorsque la valeur dudit premier registre de mémorisation (A) est inférieur à celle dudit deuxième registre de mémorisation (B).

FIG.1

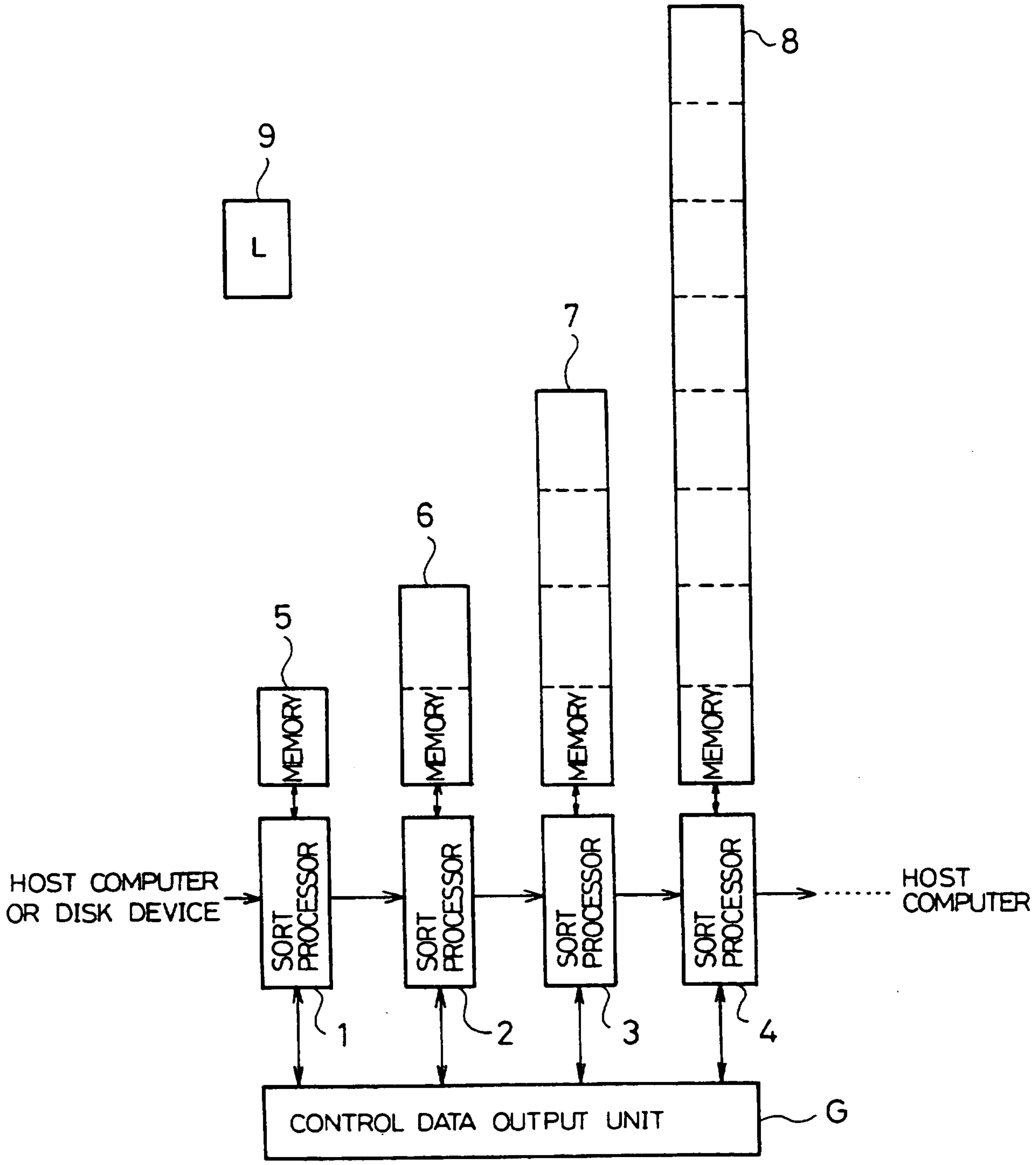

9
L
8
7
6
5
MEMORY
MEMORY
MEMORY
MEMORY
HOST COMPUTER OR DISK DEVICE
SORT PROCESSOR
SORT PROCESSOR
SORT PROCESSOR
SORT PROCESSOR
HOST COMPUTER
1
2
3
4
CONTROL DATA OUTPUT UNIT
G

FIG. 2

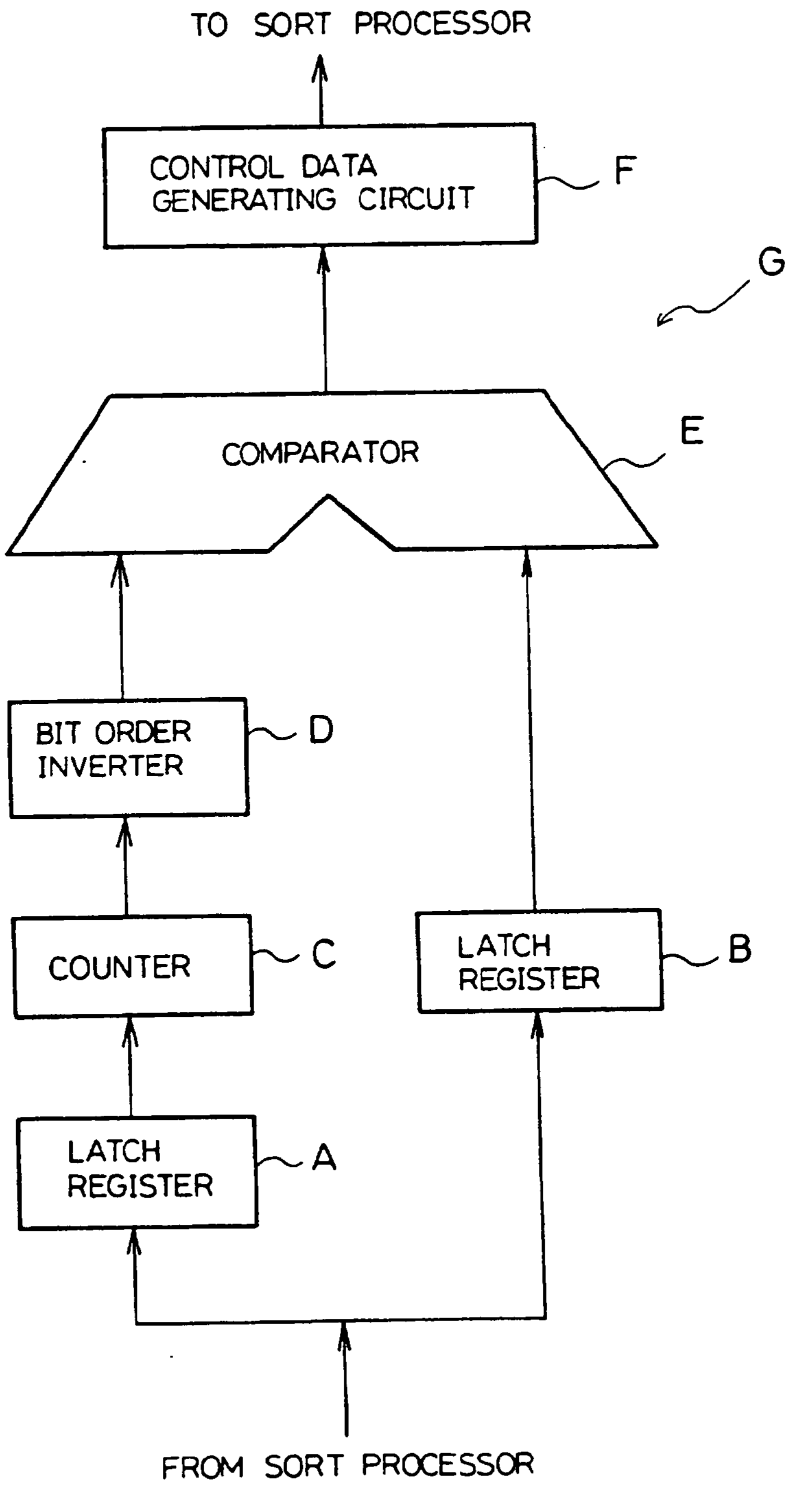
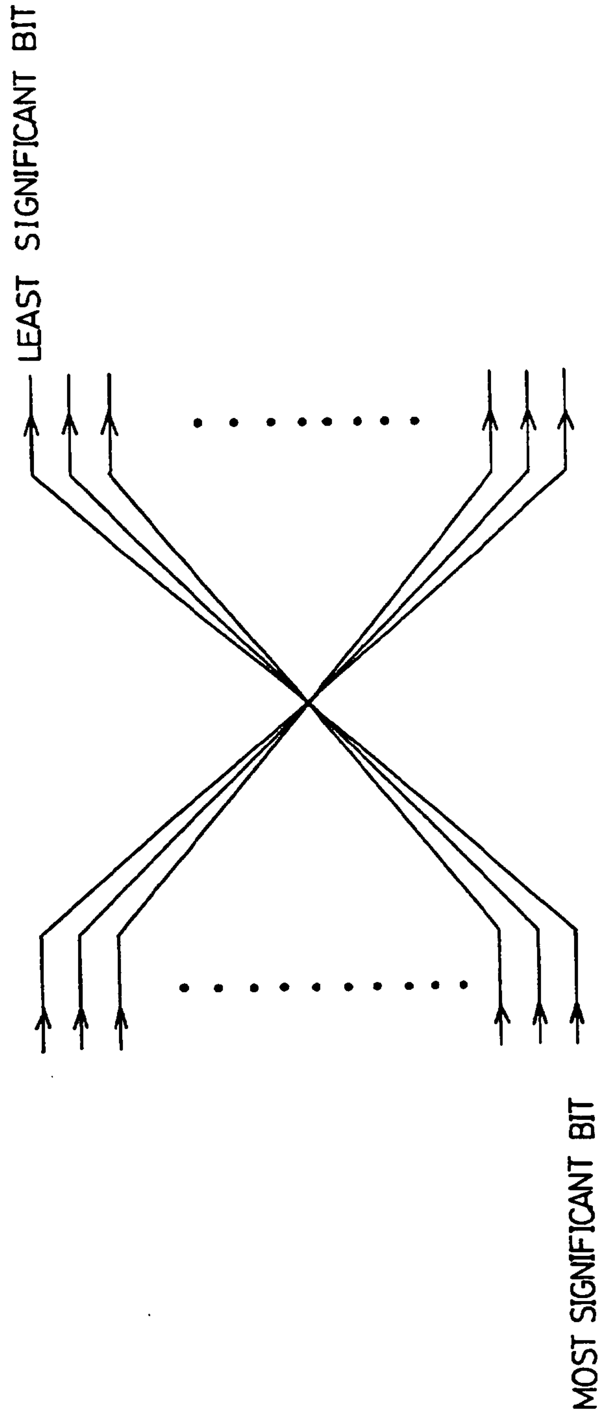

TO SORT PROCESSOR
CONTROL DATA GENERATING CIRCUIT
F
G
COMPARATOR
E
BIT ORDER INVERTER
D
COUNTER
C
LATCH REGISTER
B
LATCH REGISTER
A
FROM SORT PROCESSOR

FIG. 3



F I G. 4

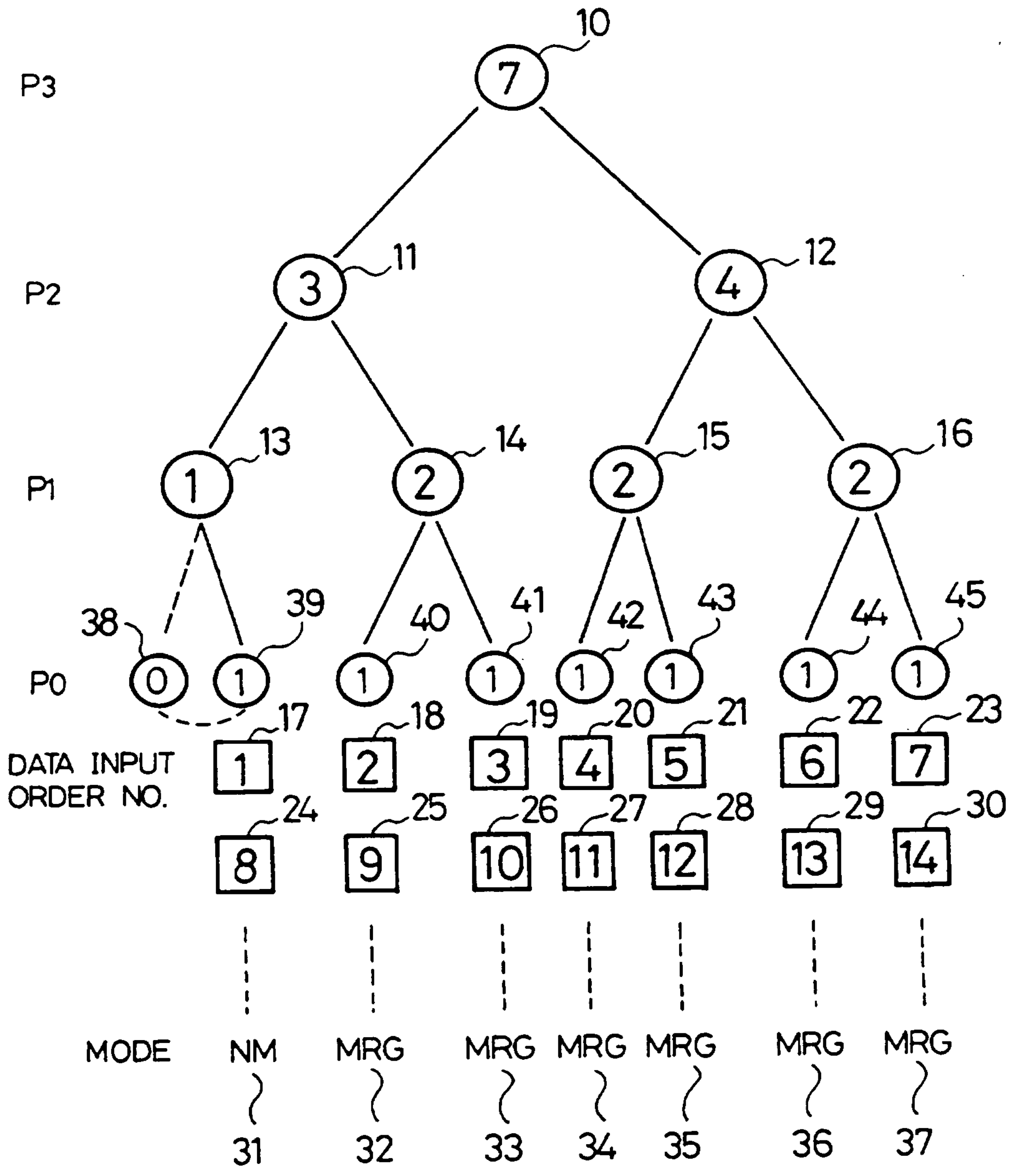

10
P3
7
11
12
P2
3
4
13
14
15
16
P1
1
2
2
2
38
39
40
41
42
43
44
45
P0
0
1
1
1
1
1
1
1
DATA INPUT ORDER NO.
17
18
19
20
21
22
23
1
2
3
4
5
6
7
24
25
26
27
28
29
30
8
9
10
11
12
13
14
MODE
NM
MRG
MRG
MRG
MRG
MRG
MRG
31
32
33
34
35
36
37

FIG. 5

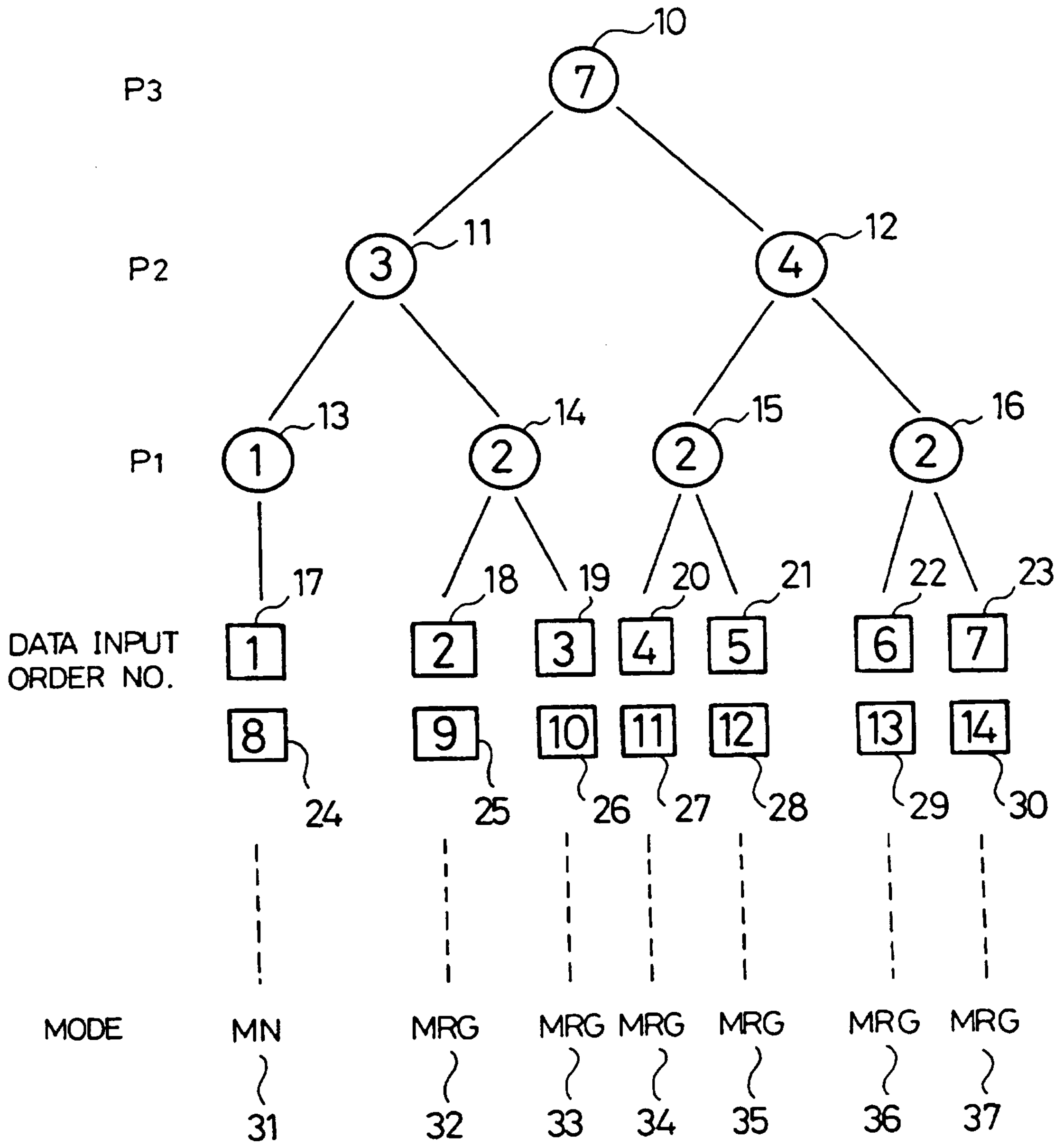
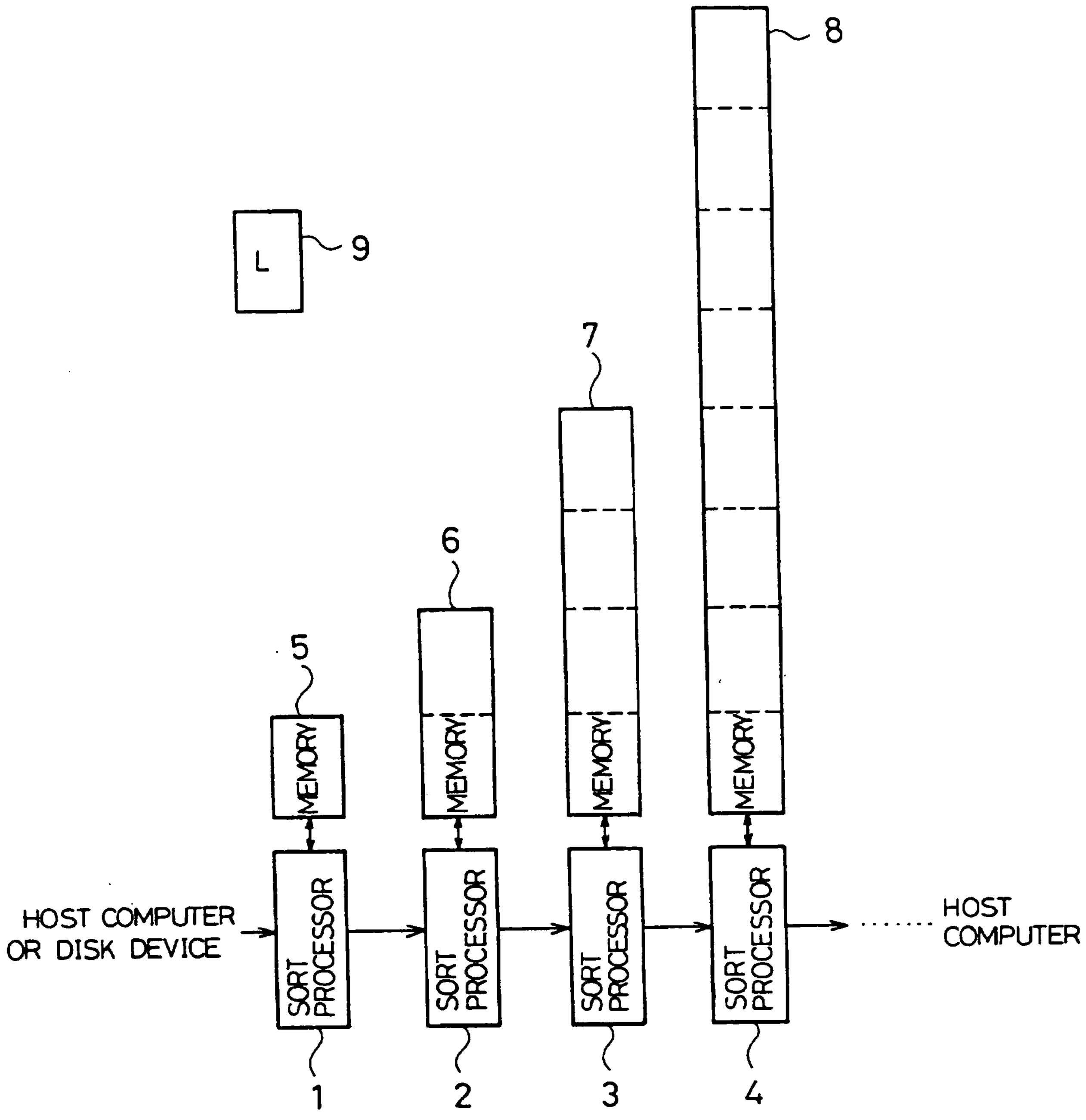

10
P3
7
11
12
P2
3
4
13
14
15
16
P1
1
2
2
2
17
18
19
20
21
22
23
DATA INPUT ORDER NO.
1
2
3
4
5
6
7
8
9
10
11
12
13
14
24
25
26
27
28
29
30
MODE
MN
MRG
MRG
MRG
MRG
MRG
MRG
31
32
33
34
35
36
37

FIG. 6 (PRIOR ART)


8
9
L
7
6
5
MEMORY
MEMORY
MEMORY
MEMORY
HOST COMPUTER
OR DISK DEVICE
SORT PROCESSOR
SORT PROCESSOR
SORT PROCESSOR
SORT PROCESSOR
HOST COMPUTER
1
2
3
4